

Langages Formels

TD 5

Guillaume Scerri
guillaume.scerri@lmf.cnrs.fr

12 mars 2026

Exercise 1: Automata \leftrightarrow MSO

1. Give closed MSO formulas recognizing :
 - (a) The language of words starting with an a . You can start by defining a formula $First(x)$ with a free variable x such that for every word w , $w, x \rightarrow i \models First(x)$ iff $i = 1$.
 - (b) The language of words in which for every a , there is a b or c further in the word.
 - (c) The language of words in which for every a , there is a b or c at the next position.
 - (d) The language of words in which the letter a can only appear in even positions.
2. Give equivalent automata to each of the following MSO formulas :
 - (a) $\exists x. First(x)$
 - (b) $\forall x. First(x)$
 - (c) $\exists x. \exists y. (y > x \wedge P_c(x))$

Exercise 2: MSO and transitive closures

Let R be a binary relation over positions of a word w which can be defined with MSO, that is such that there is an MSO formula $\varphi(x_1, x_2)$ with two free variables x_1, x_2 such that a pair of positions (i, j) is in R iff $w, x_1 \mapsto i, x_2 \mapsto j \models \varphi(x_1, x_2)$.

1. Write an MSO formula for the relation R_1 defined by : if the current letter is an a , jump two positions to the right ; if the current letter is a b , jump one position to the left ; if the current letter is a c , jump either three positions to the right. That is, given a word w , $R_1 = \{ (i, j) : w_i = a \text{ and } j = i + 2 \} \cup \{ (i, j) : w_i = b \text{ and } j = i - 1 \} \cup \{ (i, j) : w_i = c \text{ and } j = i + 3 \}$.
(You can start by defining formulas $S_k(x, y)$, resp. $P_k(x, y)$, with two free variables x, y such that for every word w , $w, x \rightarrow i, y \rightarrow j \models S_k(x, y)$, resp $P_k(x, y)$, iff $j = i + k$, resp $j = i - k$)
2. Write an MSO formula $\psi_1(x_3, x_4)$ with two free variables x_3 and x_4 defining a binary relation R_1^* which is the reflexive transitive closure of R_1 .
3. More generally, build an MSO formula defining the reflexive transitive closure of an MSO-defined binary relation.
4. For every binary MSO-definable relation R , write a closed MSO formula which is evaluated to true over a word iff the pair of the first and last positions of the word is in the reflexive transitive closure of R .
5. Give a closed MSO formula recognizing the language of words of length a multiple a three.

Exercise 3: Regular identities

We study identities on regular expressions r, s, t . Here, $r = s$ means $\mathcal{L}(r) = \mathcal{L}(s)$.

1. Prove the following identities :
 - (a) $(r + s)t = rt + st$
 - (b) $(r^*)^* = r^*$
 - (c) $(rs + r)^*r = r(sr + r)^*$
2. Prove or disprove the following identities :
 - (a) $(r + s)^* = r^* + s^*$
 - (b) $(r^*s^*)^* = (r + s)^*$
 - (c) $s(rs + s)^*r = rr^*s(rr^*s)^*$

Exercise 4: Minimal Regular expression

We define by $|e|$ the size of a regular expression e , i.e. the number of symbols appearing in e .

1. Is there a unique minimal regular expression e such that $\mathcal{L}(e) = L$ for a given regular language L ?
2. Give a procedure which, given a finite automaton \mathcal{A} , returns a minimal regular expression e such that $\mathcal{L}(\mathcal{A}) = \mathcal{L}(e)$.

Exercise 5: Résiduels

Calculer les résiduels de $\mathcal{L} = a^*(aa + b) + b(a + ba)^*$ et construire son automate minimal.

Exercise 6: Two way automata (Boustrophédon)

A two way automaton is a finite automaton which, for each transition, can move its reading head one step to the right or one step to the left (if there is nothing left to read the automaton reads a start or end marker). Equivalently, it is a Turing machine with one tape which cannot write.

1. Show that w.l.o.g. we can restrict two way automata to only accept when reading the end marker.
2. Build a two way automaton with $O(n)$ states that accept $\Sigma^*a\Sigma^n$.
3. Show that all language accepted by a deterministic two way automaton is regular (hint : you may want to use Myhill-Nerode).
4. Show that from any deterministic two way automaton with n states, we can construct an equivalent deterministic finite automaton with $2^{O(n^2)}$ states.

Exercise 7: Closure by morphism

1. Let h be the morphism $h(a) = 01$ and $h(b) = 0$. Give $h(a(a + b)^*)$.
2. Apply the construction of closure by morphism to this example.
3. Let h' be the morphism $h'(0) = ab$, $h'(1) = \varepsilon$. Give $h'^{-1}(\{ abab, baba \})$.
4. Apply the construction of closure by inverse morphism to this example.
5. Let $L = (00 \cup 1)^*$, $h(a) = 01$ and $h(b) = 10$. What is $h^{-1}(1001)$? $h^{-1}(010110)$? $h^{-1}(L)$? What is $h(h^{-1}(L))$, and is it related to L ? Apply the construction by inverse morphism to this example.
6. Let \mathcal{A} be an automaton over A , and \mathcal{B} an automaton over B . Show that for every pair (s, t) of states of \mathcal{B} and every state q of \mathcal{A} , we can compute the following set :

$$R_{s,t}(q) = \{ q' : \exists u, \exists v \in f(u), s \xrightarrow{v}_{\mathcal{B}} t, q \xrightarrow{u}_{\mathcal{A}} q' \}$$

Evaluate the complexity of your algorithm.

7. Prove that the language $\{ a^nba^n : n \geq 1 \}$ is not regular using closures and the unregularity of $\{ 0^n1^n : n \geq 1 \}$